

Weekly Report

June 18, 2017

1 Word2Vec

本周又重新看了一遍Word2Vec的具体原理，可以在Location2Vec的文章中介绍更加清楚一些。

Word2Vec是一个将文档中的单词转换为向量的算法，同时这种向量保持了单词在语义上的相似性，即相似的单词有接近的向量表示。Word2Vec建模的基本思想是令单词与其相邻的文本相似性比较高，也就是相比于无关单词，单词 w 在它的上下文 $content(w)$ 语境中出现的概率最大：

$$L = \sum_{w \in C} \log p(content(w)|w)$$

其中 C 是语料库。 $p(content(w)|w)$ 可以看作是单词和上下文单词多次的独立同分布采样：

$$p(content(w)|w) = \prod_{u \in content(w)} p(u|w)$$

其中 $p(u|w)$ 用于衡量 u 和 w 基于相似度的概率，即 u 和 w 的相似度占有其他相似度和的比例：

$$p(u|w) = \frac{v(u)^T v(w)}{\sum_{z \in C} v(z)^T v(w)}$$

由于分母要计算整个语料库，计算量太大，所以可以采用负采样的方法，即最小化当前单词和它上下文单词的距离，同时远离几个不相关的单词：

$$p(u|w) = \prod_{z \in \{u\} \cup NEG\{u\}} p(z|w)$$

定义指示函数 $L^w(x)$ ：

$$L^u(z) = \begin{cases} 1 & z = u \\ 0 & z \neq u \end{cases}$$

我们可以将 $p(z|w)$ 转为为如下形式：

$$p(z|w) = \begin{cases} \sigma(v(z)^T v(w)) & L^u(z) = 1 \\ 1 - \sigma(v(z)^T v(w)) & L^u(z) = 0 \end{cases}$$

$$p(z|w) = [\sigma(v(z)^T v(w))]^{L^u(z)} [1 - \sigma(v(z)^T v(w))]^{1-L^u(z)=0}$$

所以总体的目标函数可以推导为：

$$\begin{aligned} L &= \sum_{w \in C} \log p(\text{content}(w)|w) \\ &= \sum_{w \in C} \sum_{u \in \text{content} w} \sum_{z \in \{u\} \cup \text{NEG}\{u\}} \log p(z|w) \\ &= \sum_{w \in C} \sum_{u \in \text{content} w} \sum_{z \in \{u\} \cup \text{NEG}\{u\}} \{[L^u(z)][\log(\sigma(v(z)^T v(w)))] + [1 - L^u(z)][\log(1 - \sigma(v(z)^T v(w)))]\} \end{aligned}$$

之后对于 L 进行梯度计算，最大化 O 。

$$\text{令 } L(w, u, z) = [L^u(z)][\log(\sigma(v(z)^T v(w)))] + [1 - L^u(z)][\log(1 - \sigma(v(z)^T v(w)))]$$

$$\begin{aligned} \frac{\partial L(w, u, z)}{\partial v(z)} &= L^u(z)[1 - \sigma(v(z)^T v(w))]v(w) - [1 - L^u(z)][\sigma(v(z)^T v(w))]v(w) \\ &= [L^u(z) - \sigma(v(z)^T v(w))]v(w) \end{aligned}$$

那么对于 $v(z)$ 嵌入的向量来说，可以如下更新方式：

$$v(z) := v(z) + \eta[L^u(z) - \sigma(v(z)^T v(w))]v(w)$$

类似的，对于 $v(w)$ 来说，可以利用一下梯度进行更新

$$v(w) := v(w) + \eta[L^u(z) - \sigma(v(z)^T v(w))]v(z)$$

2 Paper Reading

2.1 Learning Person Trajectory Representations for Team Activity Analysis

本文基于球场上球员的运动轨迹对球员的行为／球队进行预测。球员的轨迹按照均匀时间采样形成具有两个通道的向量，包括x, y坐标，（类似于图像的RGB三通道）。然后使用CNN对球员轨迹进行学习。在学习过程中，将两两球员的轨迹

拼接起来，可以用于预测球员的行为。当我们把球队中所有球员的轨迹都拼接起来作为输入时，可以用于预测是哪个球队。文章中多次提到表达，我认为可以看作是特征抽取的意思，在深度学习之前都是人工进行特征的选择和训练。如今，我们通过设计深度学习的输入（轨迹数值、轨迹图片）和网络结构学习得到轨迹的表达。文章也没有介绍到底什么是表达，可能在机器学习领域表达是一种比较抽象普遍的概念（去年VAST研究神经网络的一篇文章也称深度学习是对数据的一种表达），不过最终很少有人研究表达是怎么样的，而是直接用于预测。文章说到她们的表达有两种优点：1）相比于从视频中抽取位置信息，直接使用坐标能够提供更多空间信息。2）相比于人工抽取出特征，本文使用了卷积神经网络进行学习。我认为可以参考一些关于表达的阐述和用途。

2.2 Linear Discriminative Star Coordinate for Exploring Class and Cluster Separation of High Dimensional Data

本文提出的线性投影算法基本思路是：1）对于标记数据，使同一个类的数据尽量接近，使不同类的聚类比较远。2）对于未标记数据，使用k-means对于投影后的点进行聚类，然后使用先前的方法计算，这样不断交替处理。最后可视化系统中用户可以自由选取每个维度的权重以及确认维度是否用于投影。

2.3 Discovery of Evolving Semantics through Dynamic Word Embedding Learning

前有点类似的想法：探索一个单词在不同时间下的演化（比如，Apple）。以前的工作是对每个时间段的单词分别利用词嵌入生成向量，然后不同时间段的向量进行匹配等操作。本文将匹配这个过程结合在训练过程中，目标函数增加一项 $|U_t - U_{t-1}|$ ，保证了前后的连续性。

2.4 Exploring the Context of Locations for Personalized Location Recommendations

本文使用check-in数据对用户进行位置推荐，主要是将用户和地点都嵌入到同一个高维空间中（类似于doc2vec），利用向量内积求相似性 $u_u^T v_l$ ， u_u 是用户向量， v_l 是地点向量。特别的，当考虑到时间因素的时候，给时间也一个特征向量 w_t ，用户对于一个地点在某个时间的喜爱程度决定于 $u_u^T + u_u^T w_t + v_l^T w_t$ 。